UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/669,948	09/24/2003	David Dice	6000-32500	8314

58467 7590 01/06/2010

MHKKG/SUN
P.O. BOX 398
AUSTIN, TX 78767

EXAMINER	
ANYA, CHARLES E	

ART UNIT	PAPER NUMBER
2194	

NOTIFICATION DATE	DELIVERY MODE
01/06/2010	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

patent_docketing@intprop.com
ptomhkkg@gmail.com

Office Action Summary	**Application No.** 10/669,948	**Applicant(s)** DICE ET AL.	
	Examiner CHARLES E. ANYA	**Art Unit** 2194	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3/* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *28 September 2009*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1,3-11,17-31,34-48,51,52,55-58,61,62,66,67,72 and 73* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1,3-11,17-31,34-48,51,52,55-58,61,62,66,67,72 and 73* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _____.

4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1, 3-11, 17-31, 34-48, 51, 52, 55-58, 61, 62, 66, 67 and 72-73 are pending in this application.

Claim Objections

2. Claims 36-40, 42, 46 and 72 are objected to because of the following informalities:

Page 2 (line 2) of Applicant's remark/argument includes the cancellation of claims 36-40 and 42, however pages 8 and 9 indicate otherwise (i.e. claims 36-40 and 42 are not cancelled).

Clarification is required.

Claims 46 and 72 appears to include typographical errors. Specifically, the terms "that that" on line 3 of claim 46 and "the said" on line 11 of claim 72 may have been used in error.

Appropriate correction is required.

Claim Rejections - 35 USC § 112

The following is a quotation of the second paragraph of 35 U.S.C. 112:

> **The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.**

3. Claim 44 is rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

The term “extremely” in claim 44 is a relative term which renders the claim indefinite. These terms “extremely” is not defined by the claim, and after a text of the specification for these terms, the specification does not provide a standard for ascertaining the requisite degree, and one of ordinary skill in the art would not be reasonably apprised of the scope of the invention.

For the purpose of this office action the Examiner would broadly interpret the term.

Claim Rejections - 35 USC § 112

The following is a quotation of the first paragraph of 35 U.S.C. 112:

> The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

4. Claim 28 is rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the written description requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to reasonably convey to one skilled in the relevant art that the inventor(s), at the time the application was filed, had possession of the claimed invention.

The term “revoking the bias of the biasable lock in response to detecting a given level of contention” of claim 28 is not described in the specification in such a way as to reasonably convey to one skilled in the relevant art that the inventor(s), at the time this application was filed.

5. Claims 29 and 30 are rejected for the same reason as claim 28 above.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

6. Claims 1, 3, 6, 24, 31, 35, 36, 42-44, 46-48, 52, 55, 57, 58, 61, 62, 66 and 67 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Pat. No. 6,662,364 B1 issued to Burrows et al. in view of U.S. Pat. No. 5,860126 issued to Mittal.

7. As to claim 1, Burrows teaches a computer readable storage medium storing program instructions executable by a processor of a multithreaded system to implement a mutual exclusion mechanism comprising:

a biasable lock that is concurrently accessible to a plurality of threads on the multithreaded system and is biased to at most one of the plurality of threads at a given time (“...target mutex...” Col. 2 Ln. 53 - 67, “…Mutex M…” Col. 4 Ln. 1 – 18, Ln. 56 – 67, “…Mutex M (210) Col. 5 Ln. 1 – 36);

acquisition and release sequences (“…acquire mutex M(210)…releases the mutex..." Col. 5 Ln. 1 - 14) that, when executed by a bias-holding thread on the processor, wherein the biasable lock is biased to the bias-holding thread, is free of atomic read-modify-write operations (“…fast nonatomic load/store sequence…” Col. 2 Ln. 53 – 64, “…fast nonatomic synchronization sequence…” Col. 3 Ln. 1 – 3, Col. 4 Ln. 1 – 18, Ln. 56 – 67).

Burrows is silent with reference to acquisition and release sequences that is free of memory barrier operations executable to constrain the processor from performing at least one type of instruction reordering; and wherein the processor supports out-of-order instruction execution.

Mittal teaches acquisition and release sequences that is free of memory barrier operations executable to constrain the processor from performing at least one type of instruction reordering (“…it is possible for a processor to reorder these accesses…” Col. 2 Ln. 34 – 38, “…Weak consistency is a memory access ordering model that removes an implied ordering requirements…Accesses A-D can be reordered in any manner…”

Col. 7 Ln. 10 – 39); and wherein the processor supports out-of-order instruction execution ("…it is possible for a processor to reorder these accesses…" Col. 2 Ln. 34 – 38, "…processor consistency…can be implemented simply by allowing only future reads to be out of order…" Col. 6 Ln. 48 – 67, "…processing devices (such as processors) are able to reorder access to memory…" Col. 2 Ln. 55 – 62, Col. 6 Ln. 21 – 22).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the system of Burrows with the teaching of Mittal because teaching of Mittal would improve the system of Burrows by providing a technique for controlling memory access ordering in a multi-processing system that enhances the efficiency and speed of data processing by allowing a processor to proceed or continue with the next access when a particular access is pending (Mittal Col. 1 Ln. 21 – 26).

8. As to claim 3, Burrows teaches the computer readable storage medium of claim 1, wherein the acquisition and release sequences include only read and write operations when executed by the bias-holding thread ("…fast nonatomic load/store sequence…" Col. 2 Ln. 53 – 67).

9. As to claim 6, Burrows teaches the computer readable storage medium of claim 1, wherein the biasable lock is biased on acquisition ("…the mutex is updated to indicate the new state of the mutex…" Col. 5 Ln. 29 – 35).

10. As to claim 24, Burrows teaches the computer readable storage medium of claim 1, wherein the biasable lock is rebiasable to another thread during course of a computation that employs the biasable lock (Step 280 Col. 5 Ln. 37 – 54).

11. As to claim 31, Burrows teaches a computer readable storage medium storing program instructions executable by a processor of a multithreaded system to implement a biasable lock wherein:

the biasable lock is concurrently accessible to a plurality of threads on the multithreaded system and is biased to at most one of the plurality of threads at a given time ("...target mutex..." Col. 2 Ln. 53 - 67, "…Mutex M…" Col. 4 Ln. 1 – 18, Ln. 56 – 67, "…Mutex M (210) Col. 5 Ln. 1 – 36);

the biasable lock provides at least two acquisition sequences including a fast path acquisition sequence for a bias-holding thread to which the biasable lock has been biased ("…fast nonatomic load/store sequence…" Col. 2 Ln. 53 – 64, "…fast nonatomic synchronization sequence…" Col. 3 Ln. 1 – 3, "…fast nonatomic sequence, H(M)=0…" Col. 4 Ln. 1 – 18, Ln. 56 – 67) and a second acquisition sequence, the fast path acquisition sequence optimized relative to the second acquisition sequence ("…more complex procedure for synchronization…" Col. 3 Ln. 1 – 3, "…expensive atomic hardware instructions, H(M0 = 1…" Col. 4 Ln. 1 – 4);

wherein the fast path acquisition sequence is free of atomic read-modify-write operations ("…fast nonatomic load/store sequence…" Col. 2 Ln. 53 – 64, "…fast nonatomic synchronization sequence…" Col. 3 Ln. 1 – 3, Col. 4 Ln. 1 – 18, Ln. 56 – 67).

Burrows is silent with reference to a acquisition sequence being free of explicit memory barrier operations executable to constrain the processor from performing at least one type of instruction reordering; and wherein the processor supports out-of-order instruction execution.

Mittal teaches a acquisition sequence free of explicit memory barrier operations executable to constrain the processor from performing at least one type of instruction reordering ("…it is possible for a processor to reorder these accesses…" Col. 2 Ln. 34 – 38, "…Weak consistency is a memory access ordering model that removes an implied ordering requirements…Accesses A-D can be reordered in any manner…" Col. 7 Ln. 10 – 39); and wherein the processor supports out-of-order instruction execution ("…it is possible for a processor to reorder these accesses…" Col. 2 Ln. 34 – 38, "…processor consistency…can be implemented simply by allowing only future reads to be out of order…" Col. 6 Ln. 48 – 67, "…processing devices (such as processors) are able to reorder access to memory…" Col. 2 Ln. 55 – 62, Col. 6 Ln. 21 – 22).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the system of Burrows with the teaching of Mittal because teaching of Mittal would improve the system of Burrows by providing a technique for controlling memory access ordering in a multi-processing system that enhances the efficiency and speed of data processing by allowing a processor to proceed or continue with the next access when a particular access is pending (Mittal Col. 1 Ln. 21 – 26).

12. As to claim 35, Burrows teaches computer readable storage medium of claim 31, wherein the biasable lock is rebiasable ("…Treq is associated with Mutex M…, the request is granted…" Col. 5 Ln. 37 – 42).

13. As to claim 36, Burrows teaches a computer implemented method, the method comprising:

a instantiating biasable lock that is concurrently accessible to a plurality of threads on the multithreaded system and is biased to at most one of the plurality of threads at a given time ("…target mutex..." Col. 2 Ln. 53 - 67, Col. 3 Ln. 1 – 35, Col. 1 – 18, Mutexes 128 Col. 4 Ln. 46 – 48, "…new created mutexes…" Col. 7 Ln. 41 – 52); and

for the thread to which bias has been directed, releasing and acquiring the biasable lock using fast path instruction sequences that are free of atomic read- modify-write operations ("…fast nonatomic load/store sequence…" Col. 2 Ln. 53 – 64, "…fast nonatomic synchronization sequence…" Col. 3 Ln. 1 – 3, "…fast nonatomic sequence, H(M)=0…" Col. 4 Ln. 1 – 18, Ln. 56 – 67).

Burrows is silent with reference to releasing and acquiring instruction sequence that is free of explicit memory barrier operations executable to constrain the processor from performing at least one type of instruction reordering.

Mittal teaches releasing and acquiring instruction sequence that is free of explicit memory barrier operations executable to constrain the processor from performing at least one type of instruction reordering ("…it is possible for a processor to reorder these accesses…" Col. 2 Ln. 34 – 38, "…Weak consistency is a memory access ordering model that removes an implied ordering requirements…Accesses A-D can be reordered in any manner…" Col. 7 Ln. 10 – 39).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the system of Burrows with the teaching of Mittal because teaching of Mittal would improve the system of Burrows by providing a technique for controlling memory access ordering in a multi-processing system that enhances the efficiency and speed of data processing by allowing a processor to proceed or continue with the next access when a particular access is pending (Mittal Col. 1 Ln. 21 – 26).

14. As to claim 42, Burrows teaches the method of claim 36, further comprising: rebiasing the biasable lock to another thread (Step 280 Col. 5 Ln. 37 – 54).

15. As to claim 43, Burrows teaches the method of claim 36, further comprising: executing the program code as a multi-threaded software application ("…multithreaded…" Col. 2 Ln. 30 – 49) configured to utilize the biasable lock to allow the bias-holding thread of the multithread software application to repeatedly acquire and release the lock, at least in part by executing one or more of the fast path instruction sequence ("…Treq is associated with Mutex M…, the request is granted…" Col. 5 Ln. 37 – 42).

16. As to claim 44, Burrows teaches the method of claim 43, further comprising: after rebiasing to another thread, allowing another thread to repeatedly acquire and release the lock with extremely low overhead (Step 280 Col. 5 Ln. 37 – 54).

17. As to claim 46, Burrows teaches a computer readable storage medium storing program instructions executable by a processor of a multithreaded system to implement:

a biasable lock that is concurrently accessible to a plurality of threads on the multithreaded system and is biased to at most one of the plurality of threads at a given time and maintains both a lock state ("…target mutex..." Col. 2 Ln. 53 - 67, Col. 3 Ln. 1 – 35, Col. 1 – 18, "…expensive atomic hardware instructions, H(M0 = 1…" Col. 4 Ln. 1 – 4, Mutexes 128 Col. 4 Ln. 46 – 48) and bias state ("…fast nonatomic load/store sequence…" Col. 2 Ln. 53 – 64, "…fast nonatomic synchronization sequence…" Col. 3 Ln. 1 – 3, "…fast nonatomic sequence, H(M)=0…" Col. 4 Ln. 1 – 18, Ln. 56 – 67), whereby acquisition of the lock by a thread to which bias has been directed is more efficient than acquisition of the lock by another thread ("…more complex procedure for synchronization…" Col. 3 Ln. 1 – 3, "…expensive atomic hardware instructions, H(M0 = 1…" Col. 4 Ln. 1 – 4) and does not include any atomic read-modify-write operations ("…fast nonatomic load/store sequence…" Col. 2 Ln. 53 – 64, "…fast nonatomic synchronization sequence…" Col. 3 Ln. 1 – 3, "…fast nonatomic sequence, H(M)=0…" Col. 4 Ln. 1 – 18, Ln. 56 – 67).

Burrows is silent with reference to acquisition of lock that does not include memory barrier operations executable to constrain the processor from performing at least one type of instruction reordering; and wherein the processor supports out-of-order execution.

Mittal teaches acquisition of lock that does not include memory barrier operations executable to constrain the processor from performing at least one type of instruction reordering ("…it is possible for a processor to reorder these accesses…" Col. 2 Ln. 34 – 38, "…Weak consistency is a memory access ordering model that removes an implied ordering requirements…Accesses A-D can be reordered in any manner…" Col. 7 Ln. 10 – 39); and wherein the processor supports out-of-order execution ("…it is possible for a processor to reorder these accesses…" Col. 2 Ln. 34 – 38, "…processor consistency…can be implemented simply by allowing only future reads to be out of order…" Col. 6 Ln. 48 – 67, "…processing devices (such as processors) are able to reorder access to memory…" Col. 2 Ln. 55 – 62, Col. 6 Ln. 21 – 22).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the system of Burrows with the teaching of Mittal because teaching of Mittal would improve the system of Burrows by providing a technique for controlling memory access ordering in a multi-processing system that enhances the efficiency and speed of data processing by allowing a processor to proceed or continue with the next access when a particular access is pending (Mittal Col. 1 Ln. 21 – 26).

18. As to claim 47, Burrows teaches the computer readable storage medium of claim 46, wherein the lock is rebiasable to another thread during course of a computation that employs the lock (Step 280 Col. 5 Ln. 37 – 54).

19. As to claim 48, Burrows teaches the computer readable storage medium of claim 46, including acquisition and release sequences that, when executed by the thread to which bias has been directed, include only read and write operations ("…fast nonatomic load/store sequence…" Col. 2 Ln. 53 – 67).

20. As to claim 52, Burrows teaches a computer readable storage medium storing program instructions executable by one or more processors of a multithreaded system to implement a program product including a mutual exclusion mechanism embodied therein, the program product comprising:

a data structure instantiable in memory (Disk 104/System Memory Unit 114) of a processor (Central Processing Unit 102) to implement a lock that includes a bias attribute ("…flag…" Col. 4 Ln. 46 – 48), wherein the lock is concurrently accessible to a plurality of threads on the multithreaded system and is biased to at most one of the plurality of threads at a given time ("…target mutex..." Col. 2 Ln. 53 - 67, Col. 3 Ln. 1 – 35, Col. 1 – 18, Mutexes 128 Col. 4 Ln. 46 – 48, "…new created mutexes…" Col. 7 Ln. 41 – 52);

a lock acquisition sequence of operations executable by the processor, the lock acquisition sequence having a fast path for a thread to which bias has been directed

("…fast nonatomic load/store sequence…" Col. 2 Ln. 53 – 64, "…fast nonatomic synchronization sequence…" Col. 3 Ln. 1 – 3, "…fast nonatomic sequence, H(M)=0…" Col. 4 Ln. 1 – 18, Ln. 56 – 67) and a second path ("…more complex procedure for synchronization…" Col. 3 Ln. 1 – 3, "…expensive atomic hardware instructions, H(M) = 1…" Col. 4 Ln. 1 – 4), the fast path optimized relative to the second path ("…more complex procedure for synchronization…" Col. 3 Ln. 1 – 3, "…expensive atomic hardware instructions, H(M0 = 1…" Col. 4 Ln. 1 – 4) and

wherein the fast path acquisition sequence is free of free of atomic read-modify-write operations ("…fast nonatomic load/store sequence…" Col. 2 Ln. 53 – 64, "…fast nonatomic synchronization sequence…" Col. 3 Ln. 1 – 3, "…fast nonatomic sequence, H(M)=0…" Col. 4 Ln. 1 – 18, Ln. 56 – 67).

Burrows is silent with reference to acquisition sequence is free of explicit memory barrier operations executable to constrain the processor from performing at least one type of instruction reordering; and wherein the processor supports out-of-order instruction execution.

Mittal teaches acquisition sequence is free of explicit memory barrier operations executable to constrain the processor from performing at least one type of instruction reordering ("…it is possible for a processor to reorder these accesses…" Col. 2 Ln. 34 – 38, "…Weak consistency is a memory access ordering model that removes an implied ordering requirements…Accesses A-D can be reordered in any manner…" Col. 7 Ln. 10 – 39); and wherein the processor supports out-of-order instruction execution ("…it is possible for a processor to reorder these accesses…" Col. 2 Ln. 34 – 38, "…processor consistency…can be implemented simply by allowing only future reads to be out of order…" Col. 6 Ln. 48 – 67, "…processing devices (such as processors) are able to reorder access to memory…" Col. 2 Ln. 55 – 62, Col. 6 Ln. 21 – 22).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the system of Burrows with the teaching of Mittal because teaching of Mittal would improve the system of Burrows by providing a technique for controlling memory access ordering in a multi-processing system that enhances the efficiency and speed of data processing by allowing a processor to proceed or continue with the next access when a particular access is pending (Mittal Col. 1 Ln. 21 – 26).

21. As to claim 55, Burrows teaches the computer readable storage medium of claim 52, further comprising: a lock release sequence of operations executable by the processor ("…the thread holding mutex M releases the mutex…" Col. 5 Ln. 1 – 11), the lock release sequence having a fast path for the thread to which bias has been directed ("…fast nonatomic load/store sequence…" Col. 2 Ln. 53 – 64, "…fast nonatomic synchronization sequence…" Col. 3 Ln. 1 – 3, "…fast nonatomic sequence, H(M)=0…" Col. 4 Ln. 1 – 18, Ln. 56 – 67) and a second path, the fast path optimized relative to the second path ("…more complex procedure for synchronization…" Col. 3 Ln. 1 – 3, "…expensive atomic hardware instructions, H(M) = 1…" Col. 4 Ln. 1 – 4).

22. As to claim 57, Burrows teaches the computer readable storage medium of claim 52, wherein the lock is rebiasable to another thread during course of a computation that employs the lock (Step 280 Col. 5 Ln. 37 – 54).

23. As to claim 58, Burrows teaches a computer implemented method comprising:

biasing a lock to a first thread of execution ("…fast nonatomic load/store sequence…" Col. 2 Ln. 53 – 64, "…fast nonatomic synchronization sequence…" Col. 3 Ln. 1 – 3, "…fast nonatomic sequence, H(M)=0…" Col. 4 Ln. 1 – 18, Ln. 56 – 67), wherein the biasable lock is concurrently accessible to a plurality of threads and is biased to at most one of the plurality of threads at a given time ("…target mutex..." Col. 2 Ln. 53 - 67, Col. 3 Ln. 1 – 35, Col. 1 – 18, Mutexes 128 Col. 4 Ln. 46 – 48, "…new created mutexes…" Col. 7 Ln. 41 – 52); and

subsequently acquiring the lock for, or by, the first thread with computational overhead less than for a second thread to which the lock is not currently biased ("…more complex procedure for synchronization…" Col. 3 Ln. 1 – 3, "…expensive atomic hardware instructions, H(M) = 1…" Col. 4 Ln. 1 – 4), wherein said acquiring is performed free of atomic read-modify-write operations ("…fast nonatomic load/store sequence…" Col. 2 Ln. 53 – 64, "…fast nonatomic synchronization sequence…" Col. 3 Ln. 1 – 3, "…fast nonatomic sequence, H(M)=0…" Col. 4 Ln. 1 – 18, Ln. 56 – 67).

Burrows is silent with reference to a computer processor that supports out-of-order instruction execution performing, and wherein said acquiring is performed free of explicit memory barrier operations executable to constrain the processor from performing at least one type of instruction reordering.

Mittal teaches a computer processor that supports out-of-order instruction execution performing ("…it is possible for a processor to reorder these accesses…" Col. 2 Ln. 34 – 38, "…processor consistency…can be implemented simply by allowing only future reads to be out of order…" Col. 6 Ln. 48 – 67, "…processing devices (such as processors) are able to reorder access to memory…" Col. 2 Ln. 55 – 62, Col. 6 Ln. 21 – 22), and wherein said acquiring is performed free of explicit memory barrier operations executable to constrain the processor from performing at least one type of instruction reordering ("…it is possible for a processor to reorder these accesses…" Col. 2 Ln. 34 – 38, "…Weak consistency is a memory access ordering model that removes an implied ordering requirements…Accesses A-D can be reordered in any manner…" Col. 7 Ln. 10 – 39).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the system of Burrows with the teaching of Mittal because teaching of Mittal would improve the system of Burrows by providing a technique for controlling memory access ordering in a multi-processing system that enhances the efficiency and speed of data processing by allowing a processor to proceed or continue with the next access when a particular access is pending (Mittal Col. 1 Ln. 21 – 26).

24. As to claim 61, Burrows teaches the method of claim 58, further comprising: rebiasing the lock to a third thread of execution (Step 280 Col. 5 Ln. 37 – 54).

25. As to claim 62, Burrows teaches a computer system, comprising:

a memory coupled to the processor (Disk 104/System Memory Unit 114/Central Processing Unit 102), the memory storing program instructions executable by the processor to implement:

a biasable lock ("…fast nonatomic load/store sequence…" Col. 2 Ln. 53 – 64, "…fast nonatomic synchronization sequence…" Col. 3 Ln. 1 – 3, "…fast nonatomic sequence, H(M)=0…" Col. 4 Ln. 1 – 18, Ln. 56 – 67) that is concurrently accessible to a plurality of threads on the multithreaded system and is biased to at most one of the plurality of threads at a given time ("…target mutex..." Col. 2 Ln. 53 - 67, Col. 3 Ln. 1 – 35, Col. 1 – 18, Mutexes 128 Col. 4 Ln. 46 – 48, "…new created mutexes…" Col. 7 Ln. 41 – 52) and

acquisition and release sequences that, when executed by a bias-holding thread on the processor, wherein the biasable lock is biased to the bias-holding thread, do not comprise executing atomic read-modify-write operations ("…fast nonatomic load/store sequence…" Col. 2 Ln. 53 – 64, "…fast nonatomic synchronization sequence…" Col. 3 Ln. 1 – 3, "…fast nonatomic sequence, H(M)=0…" Col. 4 Ln. 1 – 18, Ln. 56 – 67).

Burrows is silent with reference to a processor coupled to one or more other processors and configured to perform out-of-order instruction execution, and acquisition and release sequences that do not comprise explicit memory barrier operations executable to constrain the processor from performing at least one type of instruction reordering.

Mittal teaches a processor coupled to one or more other processors and configured to perform out-of-order instruction execution ("…it is possible for a processor to reorder these accesses…" Col. 2 Ln. 34 – 38, "…processor consistency…can be implemented simply by allowing only future reads to be out of order…" Col. 6 Ln. 48 – 67, "…processing devices (such as processors) are able to reorder access to memory…" Col. 2 Ln. 55 – 62, Col. 6 Ln. 21 – 22), and acquisition and release sequences that do not comprise explicit memory barrier operations executable to constrain the processor from performing at least one type of instruction reordering ("…it is possible for a processor to reorder these accesses…" Col. 2 Ln. 34 – 38, "…Weak consistency is a memory access ordering model that removes an implied ordering requirements…Accesses A-D can be reordered in any manner…" Col. 7 Ln. 10 – 39).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the system of Burrows with the teaching of Mittal because teaching of Mittal would improve the system of Burrows by providing a technique for controlling memory access ordering in a multi-processing system that enhances the efficiency and speed of data processing by allowing a processor to proceed or continue with the next access when a particular access is pending (Mittal Col. 1 Ln. 21 – 26).

26. As to claim 66, Burrows teaches the computer system of claim 62, further comprising: a rebiasing sequence, executable by the processor to bias the biasable lock to another thread on the system instead of to the bias-holding thread ("…Treq is associated with Mutex M…, the request is granted…" Col. 5 Ln. 37 – 42).

27. As to claim 67, Burrows teaches the computer system of claim 62, wherein the memory is implemented as at least one computer readable storage medium selected from the set of a disk, tape or other magnetic, optical, or electronic storage medium (Main Non-Volatile Storage Unit 102 Col. 4 Ln. 25 – 30).

28. Claims 4, 5, 7, 25, 26, 34 and 37-41 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Pat. No. 6,662,364 B1 issued to Burrows et al. in view of U.S. Pat. No. 5,860126 issued to Mittal as applied to claims 1, 31 and 36 above, and further in view of U.S. Pat. No. 6,772,153 B1 issued to Bacon et al.

29. As to claim 4, Burrows and Mittal are silent with reference to the computer readable storage medium of claim 1, wherein the biasable lock is initially unbiased.

Bacon teaches the computer readable storage medium of claim 1, wherein the biasable lock is initially unbiased ("…Data 20C…" Col. 3 Ln. 10 – 37, Col. 4 Ln. 39 – 47).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the system of Burrows and Mittal with the teaching of Bacon because teaching of Bacon would improve the system of Mittal and Burrows by providing a mechanism for assigning a new lock on request or dynamically.

30. As to claim 5, Burrows and Mittal are silent with reference to the computer readable storage medium of claim 1, wherein the biasable lock is biased on creation.

Bacon teaches the computer readable storage medium of claim 1, wherein the biasable lock is biased on creation ("…any thread 6 can obtain a new system lock by calling a function createSystemLock…" Col. 4 Ln. 25 – 31, Step 502 Col. 6 Ln. 1 – 9).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the system of Burrows and Mittal with the teaching of Bacon because teaching of Bacon would improve the system of Mittal and Burrows by providing a function mechanism for obtaining a new lock when needed (Bacon Col. 4 Ln. 25 – 28).

31. As to claim 7, Bacon teaches the computer readable storage medium of claim 1, wherein the bias-holding thread is not the creating thread ("…createSystemLock…" Col. 4 Ln. 25 – 31).

32. As to claim 25, Bacon teaches the computer readable storage medium of claim 1, wherein bias of the lock to the bias-holding thread is revocable and revocation of bias by, or on behalf of, a contending thread, is mediated, at least in part, using a signal handler ("…Locked Flag 20B is set to zero…" Col. 3 Ln. 1 – 12).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the system of Burrows and Mittal with the teaching of Bacon because teaching of Bacon would improve the system of Burrows and Mittal by allowing for previously granted permission to be revised to allow other process to be fairly granted permission, thus providing a more efficient processing.

33. As to claim 26, Bacon teaches the computer readable storage medium of claim 1, wherein bias of the lock to the bias-holding thread is revocable and revocation of bias by, or on behalf of, a contending thread, is mediated, at least in part, using a cross-call ("…systemRelease…" Col. 4 Ln. 15 – 18).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the system of Burrows and Mittal with the teaching of Bacon because teaching of Bacon would improve the system of Burrows and Mittal by allowing for previously granted permission to be revised to allow other process to be fairly granted permission, thus providing a more efficient processing.

34. As to claim 34, Bacon teaches the computer readable storage medium of claim 31, wherein the second acquisition sequence implements one of an MCS lock, a TATAS lock, a lock consistent with that provided by a POSIX pthread Mutex library and a Java monitor ("…POSIX…" Col. 4 Ln. 25 – 37).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the system of Burrows and Mittal with the teaching of Bacon because teaching of Bacon would improve the system of Burrows and Mittal by providing application programming interface (API), along with shell and utilities interfaces for software compatible with variants of Unix operating system, thus providing portability, multi-tasking and multi-user in a time-sharing configuration associated with Unix operating system.

35. As to claim 37, Bacon teaches the method of claim 36, further comprising: directing the bias to the thread coincident with a first acquisition of the biasable lock ("…any thread 6 can obtain a new system lock by calling a function createSystemLock…" Col. 4 Ln. 25 – 31, Step 502 Col. 6 Ln. 1 – 9).

36. As to claim 38, Bacon teaches the method of claim 36, further comprising: directing the bias to the thread coincident with the instantiation ("…any thread 6 can obtain a new system lock by calling a function createSystemLock…" Col. 4 Ln. 25 – 31, Step 502 Col. 6 Ln. 1 – 9).

37. As to claim 39, Bacon teaches the method of claim 36, further comprising: directing the bias to the thread coincident with creation of an object ("…When a thread T creates an object O, the system sets O to be unlocked, non-shared, and owned by T…" Col. 4 Ln. 60 – 67).

38. As to claim 40, Bacon teaches the method of claim 36, wherein directing of bias to the thread is performed by another thread ("…createSystemLock…" Col. 4 Ln. 25 – 31).

39. As to claim 41, Burrows teaches the method of claim 36, further comprising: for a thread other than the thread to which bias has been directed, acquiring the biasable lock using an instruction sequence that unbiases the lock, if then biased ("…more complex procedure for synchronization…" Col. 3 Ln. 1 – 3, "…or by conventional expensive atomic hardware instructions, H(M) = 1…" Col. 4 Ln. 1 – 4).

40. Claims 8, 11, 17, 18, 51 and 56 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Pat. No. 6,662,364 B1 issued to Burrows et al. in view of U.S. Pat. No. 5,860126 issued to Mittal as applied to claims 1, 48 and 52 above and further in view of U.S. Pat. No. 6,430,649 B1 issued to Chaudhry et al.

41. As to claim 8, Burrows and Mittal are silent with reference to the computer readable storage medium of claim 1, wherein execution of the acquisition sequence by the bias-holding thread employs a programming construct that precludes reordering of a particular read before a particular write.

Chaudhry teaches the computer readable storage medium of claim 1 wherein execution of the acquisition sequence by the bias-holding thread employs a programming construct that precludes reordering of a particular read before a particular write ("…ensure that a write operation occurs before a read operation…" Col. 8 Ln. 60 – 67, Col. 9 Ln. 1 – 12, Col. 9 Ln. 53 – 67).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the system of Burrows and Mittal with the teaching of

Chaudhry because teaching of Chaudhry would improve the system of Burrows and Mittal by ensuring that a particular read operation does not overtake a preceding write operation by flushing a write buffer in a load store unit before the read operation takes place including enforcing dependencies between memory references without incurring the delays inherent in member operations (Chaudhry Col. 1 Ln. 50 – 57, Col. 2 Ln. 1 – 3).

42. As to claim 11, Chaudhry teaches the computer readable storage medium of claim 8, wherein the programming construct employs collocation of the target of the particular read and the target of the particular write ("…ensure that a write operation occurs before a read operation…" Col. 8 Ln. 60 – 67, Col. 9 Ln. 1 – 12, Col. 9 Ln. 53 – 67).

43. As to claim 17, Chaudhry teaches the computer readable storage medium of claim 8, wherein the particular read loads lock status (Load Buffer 114 Col. 9 Ln. 1 – 12, Ln. 53 – 67); and wherein the particular write stores a quick lock indication (Store Buffer 116 Col. 9 Ln. 1 – 12, Ln. 53 – 67).

44. As to claim 18, Chaudhry teaches the computer readable storage medium of claim 8, wherein the preclusion is based, at least in part, on characteristics of an implementation of a memory model (Head Thread 202/Speculative Thread 302 Col. 8 Ln. 60 – 67, Col. 9 Ln. 1 – 12).

45. As to claim 51, Chaudhry teaches the computer readable storage medium of claim 48, wherein the acquisition sequence employs a programming construct that precludes reordering of a particular read before a particular write ("…ensure that a write operation occurs before a read operation…" Col. 8 Ln. 60 – 67, Col. 9 Ln. 1 – 12, Col. 9 Ln. 53 – 67).

46. As to clam 56, Chaudhry teaches the computer readable storage medium of claim 52, wherein the acquisition sequence employs a programming construct that precludes reordering of a particular read before a particular write ("…ensure that a write operation occurs before a read operation…" Col. 8 Ln. 60 – 67, Col. 9 Ln. 1 – 12, Col. 9 Ln. 53 – 67).

47. Claims 9 and 10 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Pat. No. 6,662,364 B1 issued to Burrows et al. in view of U.S. Pat. No. 5,860126 issued to Mittal and further in view of U.S. Pat. No. 6,430,649 B1 issued to Chaudhry et al. as applied to claim 8 above, and further in view of U.S. Pat. No. 7,398,376 B2 issued to Mckenney.

48. As to claim 9, Burrows, Mittal and Chaudhry are silent with reference to the computer readable storage medium of claim 8, wherein the precluded reordering includes reordering by a compiler.

McKenney teaches the computer readable storage medium of claim 8, wherein the precluded reordering includes reordering by a compiler ("…compiler directives…" Abstract).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the system of Burrows, Mittal and Chaudhry with the teaching of McKenney because teaching of McKenney would improve the system of Burrows, Mittal and Chaudhry by providing an execution of instruction technique for maximizing processor performance by placing constraints on shared memory access while removing constraints on non-shared memory accesses (McKenney Col. 3 Ln. 35 – 42).

49. As to claim 10, McKenney teaches the computer readable storage medium of claim 8, wherein the precluded reordering includes reordering by the processor ("…CPU…" Col. 5 Ln. 1 – 3).

50. Claim 19 is rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Pat. No. 6,662,364 B1 issued to Burrows et al. in view of U.S. Pat. No. 5,860126 issued to Mittal as applied to claim 1 above and further in view of U.S. Pat. No. 6,430,649 B1 issued to Chaudhry et al., and further in view of U.S. Pat. No. 6,678,807 B2 issued to Boatright et al.

51. As to claim 19, Chaudhry teaches the computer readable storage medium of claim 1, wherein the acquisition by the bias-holding lock includes a store operation followed in program order by a load operation ("…ensure that a write operation occurs before a read operation…" Col. 8 Ln. 62 – 67, Col. 9 Ln. 53 – 59), and Burrows teaches the store and load operations having collocated targets that encode a quick lock indication ("…flag…" Col. 4 Ln. 46 – 48) and lock status, respectively ("…H(M)=0…H(M)=1…" Col. 4 Ln. 1 – 4).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the system of Burrows and Mittal with the teaching of Chaudhry because teaching of Chaudhry would improve the system of Burrows and Mittal by ensuring that a particular read operation does not overtake a preceding write operation by flushing a write buffer in a load store unit before the read operation takes place including enforcing dependencies between memory references without incurring the delays inherent in member operations (Chaudhry Col. 1 Ln. 50 – 57, Col. 2 Ln. 1 – 3).

Burrows, Mittal and Chaudhry are silent with reference to the processor that implements a total store order (TSO) memory model.

Boatright teaches the processor that implements a total store order (TSO) memory model ("…TSO memory model…" Col. 2 Ln. 18 – 38).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the system of Burrows, Mittal and Chaudhry with the teaching of Boatright because teaching of Boatright would improve the system of

Chaudhry, Mittal and Burrows by providing a mechanism for allowing load operations that are completely covered by two or more store operations to receive data via store buffer forwarding in such a manner as to retain the side effects of the two Total Store Order (TSO) restrictions thereby increasing processor performance without violating the restrictive memory model (Boatright Col. 2 Ln. 7 – 13).

52. Claims 20 and 21 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Pat. No. 6,662,364 B1 issued to Burrows et al. in view of U.S. Pat. No. 5,860126 issued to Mittal as applied to claim 1 above, and further in view of U.S. Pat. No. 6,965,961 issued to Scott.

53. As to claim 20, Burrows and Mittal are silent with reference to the computer readable storage medium of claim 1, wherein the biasable lock includes an MCS lock augmented to provide fast path acquisition and release sequences for the thread.

Scott teaches the computer readable storage medium of claim 1, wherein the biasable lock includes an MCS lock augmented to provide fast path acquisition and release sequences for the thread ("…MCS…" Col. 12 Ln. 15 – 23).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the system of Burrows and Mittal with the teaching of Scott because teaching of Scott would improve the system of Burrows and Mittal by providing a method of implementing mutual exclusion that is fast, scalable, fair and avoids network contention.

54. As to claim 21, Scott teaches the computer readable storage medium of claim 1, wherein the biasable lock includes an TATAS lock augmented to provide fast path acquisition and release sequences for the thread ("…TASB…" Col. 12 Ln. 3 – 23).

55. Claim 22 is rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Pat. No. 6,662,364 B1 issued to Burrows et al. in view of U.S. Pat. No. 5,860126 issued to Mittal as applied to claim 1 above, and further in view of U.S. Pat. No. 6,112,222 issued to Govindaraju et al.

56. As to claim 22, Burrows and Mittal are silent with reference to the computer readable storage medium of claim 1, wherein the biasable lock includes a lock provided by a POSIX pthreads mutex library, augmented to provide fast path acquisition and release sequences for the thread.

Govindaraju teaches the computer readable storage medium of claim 1, wherein the biasable lock includes a lock provided by a POSIX pthreads mutex library ("…second lock process employing at least one function in the POSIX treads standard…" Col. 2 Ln. 51 – 65), augmented to provide fast path acquisition and release sequences for the thread ("…the lock is claimed by the thread and ownership is set…" Col. 3 Ln. 15 – 32).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the system of Burrows and Mittal with the teaching of

Govindaraju because teaching of Govindaraju would improve the system of Burrows and Mittal by providing POSIX functions for enhancing portability of applications running across multiple operating systems (Govindaraju Col. 3 Ln. 39 – 41).

57. Claim 23 is rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Pat. No. 6,662,364 B1 issued to Burrows et al. in view of U.S. Pat. No. 5,860126 issued to Mittal as applied to claim 1 above, and further in view of U.S. Pat. No. 6,757,891 B1 issued to Azagury et al.

58. As to claim 23, Burrows and Mittal are silent with reference to the computer readable storage medium of claim 1, wherein the biasable lock includes a monitor provided by a Java virtual machine implementation ("…monitors…" Col. 3 Ln. 32 – 45, "…monitor enter and monitor exit…" Col. 4 Ln. 26 – 49), the monitor augmented to provide fast path acquisition and release sequences for the thread to which bias has been directed.

Azagury teaches the computer readable storage medium of claim 1, wherein the biasable lock includes a monitor provided by a Java virtual machine implementation ("…monitors…" Col. 3 Ln. 32 – 45, "…monitor enter and monitor exit…" Col. 4 Ln. 26 – 49), the monitor augmented to provide fast path acquisition and release sequences for the thread to which bias has been directed ("…monitor enter without synchronization…monitor exit without synchronization…" Col. 4 Ln. 26 – 49).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the system of Burrows and Mittal with the teaching of Azagury because teaching of Azagury would improve the system of Burrows and Mittal by ensuring correct program operation on a multi-processor that is not sequentially consistent in order to avoid unnecessary memory synchronization costs (Azabury Col. 7 Ln. 54 – 64).

59. Claim 27 is rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Pat. No. 6,662,364 B1 issued to Burrows et al. in view of U.S. Pat. No. 5,860126 issued to Mittal as applied to claim 1 above, and further in view of U.S. Pat. No. 6,081,665 A issued to Nilsen et al.

60. As to claim 27, Burrows and Mittal are silent with reference to the computer readable storage medium of claim 1, wherein bias of the lock to the bias-holding thread is revocable and revocation of bias by, or on behalf of, a contending thread, is handled, at least in part, at a garbage collection safe point.

Nilsen teaches the computer readable storage medium of claim 1, wherein bias of the lock to the bias-holding thread is revocable and revocation of bias by, or on behalf of, a contending thread, is handled, at least in part, at a garbage collection safe point ("…point where safe garbage collection can take place…" Col. 66 Ln. 63 – 67).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the system of Burrows and Mittal with the teaching of

Nilsen because teaching of Nilsen would improve the system of Burrows and Mittal by providing runtime guarantees that preempted or suspended threads will not make any further progress in execution or use of system resources.

61. Claims 28-30 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Pat. No. 6,662,364 B1 issued to Burrows et al. in view of U.S. Pat. No. 5,860126 issued to Mittal as applied to claims 1 and 24 above, and further in view of U.S. Pub. No. 2001/0014905 A1 to Onodera.

62. As to claim 28, Burrows and Mittal are silent with reference to the computer readable storage medium of claim 24, wherein the program instructions are further executable to implement:

detecting a current level of contention;

revoking the bias of the bias able lock in response to detecting a given level of contention; and

a rebiasing sequence executable by the processor to rebias the lock to the thread in response to detecting a lower level of contention.

Onodera teaches the computer readable storage medium of claim 24, wherein the program instructions are further executable to implement:

detecting a current level of contention ("…determining whether the contention bit has been set..." page 6 paragraph 0051, "…queue of the a thread…" page 6 paragraph

0056, "…determining whether a thread that is stored in a queue is present…" page 6 paragraph 0056);

revoking the bias of the bias able lock in response to detecting a given level of contention (permitting the first thread to exit the exclusive control state…" page 6 paragraph 0053, "…permitting the first thread to exit the notification state…" page 6 paragraph 0056); and

a rebiasing sequence executable by the processor to rebias the lock to the thread in response to detecting a lower level of contention ("…terminating an unlocking process if the contention bit has not been set…" page 6 paragraph 0051, "…When the contention bit is not set, the unlocking process is terminated…" page 9 paragraph 0099).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the system of Burrows and Mittal with the teaching of Govindaraju because teaching of Govindaraju would improve the system of Burrows and Mittal by providing unlocking process that allows expensive memory synchronization commands usage to be minimized (Onodera page 6 paragraph 0052).

63. As to claim 29, Onodera teaches the computer readable storage medium of claim 28, wherein the detecting a current level of contention detection comprises accessing a request queue to identify the absence of contention "…queue of a thread…" page 6 paragraph 0056, "…determining whether a thread that is stored in a queue is present…" page 6 paragraph 0056).

64. As to claim 30, Onodera teaches the computer readable storage medium of claim 28, wherein the detecting a current level of contention detection comprises employing an attempt counter to identify the absence of contention ("…determining whether the contention bit has been set..." page 6 paragraph 0051).

65. Claims 45, 72 and 73 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Pat. No. 6,662,364 B1 issued to Burrows et al. in view of U.S. Pat. No. 5,860126 issued to Mittal as applied to claim 36 above, and further in view of U.S. Pat. No. 6,792,601 B1 issued to Dimpsey et al.

66. As to claim 45, Burrows and Mittal are silent with reference to the method of claim 36, wherein the method is performed as part executing program code in a single-threaded execution environment, wherein the program code is compiled with the biasable lock for execution on both the single-threaded execution environment and on a multi-threaded execution environment, and wherein the biasable lock allows the program code to run in the single-threaded execution environment without significant lock-related overhead.

Dimpsey teaches the method of claim 36, wherein the method is performed as part executing program executing program code in a single-threaded execution environment ("…first mode…" Col. 6 Ln. 40 – 50, Col. 7 Ln. 26 – 32, Col. 9 Ln. 31 – 39), wherein the program code is compiled with the biasable lock for execution on both the single-threaded execution environment and on a multi-threaded execution environment ("…first mode…second mode…" Col. 6 Ln. 40 – 50, Col. 9 Ln. 31 – 39: NOTE: the threads implemented in the first and second modes are Java threads and Java threads/applications are typically compliable before execution), and wherein the lock allows the program code to run in the single-threaded execution environment without significant lock-related overhead ("…object locking system…" Col. 6 Ln. 40 – 47, Col. 7 Ln. 25 – 32).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the system of Burrows and Mittal with the teaching of Dimpsey because teaching of Dimpsey would improve the system of Burrows and Mittal by providing a mechanism for providing a fast and improved performance by allowing a single thread to use all machine resources.

67. As to claim 72, Burrows teaches a computer implemented method of making a single computer program product suitable for efficient execution as a multi-threaded computation ("…multithreaded environment…" Col. 2 Ln. 45 – 50), the method comprising:

structuring a computation as a multithread computation ("…multithreaded environment…" Col. 2 Ln. 45 – 50);

mediating at least some sources of contention in the multithreaded computation using a biasable locking mechanism, wherein the bias able locking mechanism is concurrently accessible to a plurality of threads on the one or more processors and is biased to at most one of the plurality of threads at a given time ("...target mutex..." Col. 2 Ln. 53 - 67, "…Mutex M…" Col. 4 Ln. 1 – 18, Ln. 56 – 67, "…Mutex M (210) Col. 5 Ln. 1 – 36);

wherein the mediating includes providing at least two acquisition sequences including a fast path acquisition sequence for a bias-holding thread to which the biasable lock has been biased ("…fast nonatomic load/store sequence…" Col. 2 Ln. 53 – 64, "…fast nonatomic synchronization sequence…" Col. 3 Ln. 1 – 3, "…fast nonatomic sequence, H(M)=0…" Col. 4 Ln. 1 – 18, Ln. 56 – 67) and a second acquisition sequence, the fast path acquisition sequence being optimized relative to the second acquisition sequence ("…more complex procedure for synchronization…" Col. 3 Ln. 1 – 3, "…expensive atomic hardware instructions, H(M) = 1…" Col. 4 Ln. 1 – 4); and

introducing the instances of the biasable locking mechanism into program code ("…designate a thread that is currently associated with the mutex…" Col. 2 Ln. 53 – 67, "…the mutex is to be acquired exclusively, or almost exclusively by one thread…" Col. 4 Ln. 7 – 9, Ln. 60 – 61).

Burrows is silent with reference to reference to one or more processors capable of out-of-order execution, making a single computer program product suitable for efficient execution as a single-threaded computation; compiling the program code, and

encoding the compiled program code, including the instances of the bias able locking mechanism, in a computer program product.

Mittal teaches one or more processors capable of out-of-order execution ("…it is possible for a processor to reorder these accesses…" Col. 2 Ln. 34 – 38, "…processor consistency…can be implemented simply by allowing only future reads to be out of order…" Col. 6 Ln. 48 – 67, "…processing devices (such as processors) are able to reorder access to memory…" Col. 2 Ln. 55 – 62, Col. 6 Ln. 21 – 22).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the system of Burrows with the teaching of Mittal because teaching of Mittal would improve the system of Burrows by providing a technique for controlling memory access ordering in a multi-processing system that enhances the efficiency and speed of data processing by allowing a processor to proceed or continue with the next access when a particular access is pending (Mittal Col. 1 Ln. 21 – 26).

Dimpsey teaches making a single computer program product suitable for efficient execution as a single-threaded computation ("…first mode…" Col. 6 Ln. 40 – 50, Col. 7 Ln. 26 – 33, Col. 9 Ln. 31 – 39);

compiling the program code (Java Application 50 Col. 9 Ln. 31 – 39: NOTE: the threads implemented in the first and second modes are Java threads and Java threads/applications are typically compliable before execution); and

encoding the compiled program code, including the instances of the biasable locking mechanism (Java Application 50 Col. 9 Ln. 31 – 39), in a computer program product (memory 60 Col. 9 Ln. 8 – 20).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the system of Burrows and Mittal with the teaching of Dimpsey because teaching of Dimpsey would improve the system of Burrows and Mittal by providing a mechanism for providing a fast and improved performance by allowing a single thread to use all machine resources.

68. As to claim 73, Dimpsey teaches the method of claim 72, wherein the encoding includes transferring the compiled program code onto at least one computer readable storage medium selected from the set of a disk, tape or other magnetic, optical, or electronic storage medium (memory 60 Col. 9 Ln. 8 – 20).

Response to Arguments

Applicant's arguments with respect to claims 1, 3-11, 17-31, 34-48, 51, 52, 55-58, 61, 62, 66, 67 and 72-73 have been considered but are moot in view of the new ground(s) of rejection.

Conclusion

Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to CHARLES E. ANYA whose telephone number is (571)272-3757. The examiner can normally be reached on 8:30-5:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Hyung Sough can be reached on 571-272-6799. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Hyung S. Sough/
Supervisory Patent Examiner, Art Unit 2194
01/03/10

cea.